Evolution of Standards in an ~AI World

Use of LLMs to Transform IETF Standards into Rules-as-Data Form for Dynamic Discovery and Conformance Checking During Development Unit Tests

Joseph Potvin

Invited Brief to the Internet Research Task Force (IRTF)
Open Meeting at IETF-124 in Montreal 2025-11-05

IRTF

Xalgorithms Foundation

THE OPMAN COMPANY

Université du Québec en Outaouais

Apparent Intelligence

Evolution of Standards in an ~AI World

with **RuleMaker** editing app

Use of LLMs to Transform IETF Standards into Rules-as-Data Form for Dynamic Discovery and Conformance Checking During Development Unit Tests

from **RuleReserve** nodes on IPFS or your own network using Websocket + soon MCP

with **RuleTaker** functions in your programming environment using Websocket and soon, MCP

Joseph Potvin

Invited Brief to the Internet Research Task Force (IRTF)
Open Meeting at IETF-124 in Montreal 2025-11-05

IRTF

Xalgorithms Foundation

DWDS Design & Spec by: Joseph Potvin.
Reference Implementation by:
Don Kelly (Rust), Huda Hussain (SvelteJS),
Early Prototyping by: Ryan Fleck and Ted Kim.
Tabular Rules-as-Data Structuring guided by:
Bill Olders, Wayne Cunneyworth
Nhamo Mtetwa, and Dave Thomas.
Research Support by Stéphane Gagnon.

Developers of the Internet core work within a rule set involving at least 200,000 discrete rules.

There are at least 5,000 RFCs 'in effect' (non-obsoleted), and inderdependencies.

Assume ~40 discrete rules per RFC, some fewer, but others with hundreds.

A discrete rule is:
Input Conditions paired with normative Output Assertions using BCP-14 terms:
MUST / MUST NOT
SHOULD / SHOULD NOT
MAY / MAY NOT

IS + RULE ⟹ OUGHT

Adapted from the original 1953 cartoon by Alex Graham in *The New Yorker* magazine.

e.g. BGP-4 Dissemination Rule

IETF RFC specifications are published as precise prose:
- rich in meaning when read;
- poor in utility when coding.

Long-form does not scale well.

Often challenging to comprehend.

Very difficult to recall exactly.

Unit tests for conformance validation involve a lot of work.

IRFC 4271: A Border Gateway Protocol 4 (BGP-4)

...

9.1.3. Phase 3: Route Dissemination

The Phase 3 decision function is invoked on completion of Phase 2, or when any of the following events occur:

a) when routes in the Loc-RIB to local destinations have changed

b) when locally generated routes learned by means outside of BGP have changed

c) when a new BGP speaker connection has been established

The Phase 3 function is a separate process that completes when it has no further work to do. The Phase 3 Routing Decision function is blocked from running while the Phase 2 decision function is in process.

All routes in the Loc-RIB are processed into Adj-RIBs-Out according to configured policy. This policy MAY exclude a route in the Loc-RIB from being installed in a particular Adj-RIB-Out. A route SHALL NOT be installed in the Adj-Rib-Out unless the destination, and NEXT_HOP described by this route, may be forwarded appropriately by the Routing Table. If a route in Loc-RIB is excluded from a particular Adj-RIB-Out, the previously advertised route in that Adj-RIB-Out MUST be withdrawn from service by means of an UPDATE message.

...

Wouldn’t it be nice if we had a way to...

Transform 200,000+ IETF rules from thousands of RFCs into a library of consistent data structures so that they can be dynamically discovered and run (without injection risk) during unit tests, from within any programming environment for comprensible, fast and efficient conformance validation.

Finite Grammar (unlimited Semantics, but possibly a controlled vocabulary)

- Determiner: the; this; these; a
- Noun: word or phrase **(required)**
- Attribute: word or phrase **(required)**
- Description: adjective, adverb or arithmetic **(required)**
- Past Participle Verb: word or phrase referring to the subject noun
- Predicate Verb: usually a form of: to be; to do; to have

"Minimal Recursion Semantics" (MRS) for computational grammar decomposition introduced by Ann Copestake, and a grammar structure inspired by Backus Normal Form by John Backus, following from Noam Chomskys earliest work.

Backus, J. W., Bauer, F. L., Green, J., Naur, P., Rutishauser, H., Mccarthy, J., Perlis, A. J., Vauquois, B., Wegstein, J. H., Wijngaarden, A. V., & Woodger, M. (1960). Report on the Algorithmic Language ALGOL 60. Communications of the ACM, 3(5), 16. See: Backus Normal Form (BNF)

Chomsky, N. (1957). Syntactic Structures (2nd Edition, 2000). Mouton de Gruyter.

Chomsky, N. (1958). Finite State Languages. Information and Control, 1, 91–112.

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, *3*(2–3), 281–332. https://doi.org/10.1007/s11168-006-6327-9

Bisemantic four-value logic

Building on work by
Nuer Belnap and Graham Priest.

Belnap, N. D. (1977a). A Useful Four-Valued Logic. In J. M. Dunn & G. Epstein (Eds.), Modern Uses of Multiple-Valued Logic (pp. 5–37). Springer Netherlands. https://doi.org/10.1007/978-94-010-1161-7_2

Belnap, N. D. (1977b). How a Computer Should Think. In H. Omori & H. Wansing (Eds.), New Essays on Belnap-Dunn Logic. 2020. (pp. 35–53). Springer International Publishing. https://doi.org/10.1007/978-3-030-31136-0_4

Priest, G. (2014). Beyond True and False. Aeon. https://aeon.co/essays/the-logic-of-buddhist-philosophy-goes-beyond-simple-truth

Priest, G. (2018). The fifth corner of four: An essay on buddhist metaphysics and the catuṣkoṭi (First edition). Oxford University Press.

Semantic Assignment of Numeric Tetranary Elements for DWDS Logic Gate States

Binary	Symbol	Input Conditions	Output Assertions
00	00	NO	NOT
01	01	YES	MUST
10	10	YES AND NO (BOTH)	MAY
11	11	YES OR NO (UNSURE)	SHOULD

Semantic Assignment of Symbolic Tetranary Elements for DWDS Logic Gate States

Binary	Input Symbol	Input Conditions	Output Symbol	Output Assertions
00	X	NO	⊘	NOT
01	✓	YES	!!	MUST
10	&	YES AND NO	○	MAY
11	?	YES OR NO	!	SHOULD

e.g. BGP-4 Dissemination Rule

Items per page: 8 | 1–3 of 3 items

Scenarios

Input Conditions

1. This route is received in Adj-RIB-In from an eBGP peer.
2. This route is received in Adj-RIB-In from a local service.
3. This route is received in Adj-RIB-In from an iBGP peer.

Input +

Output Assertions

i. This route installed in Adj-RIB-In shall be propagated as an outbound update to all eBGP peers.

ii. This route installed in Adj-RIB-In shall be propagated as an outbound update to all iBGP peers.

Output +

DWDS
RuleMaker
Authoring app.

e.g. BGP-4 Dissemination Rule

I instructed https://venice.ai LLM to express the rule in "simple discrete declarative sentences using the finite grammar of the **Data With Direction Specification (DWDS)**"

LLM used as a linguistic aid, not an oracle

Input Conditions

1. This route is received in Adj-RIB-In from an eBGP peer.
2. This route is received in Adj-RIB-In from a local service.
3. This route is received in Adj-RIB-In from an iBGP peer.

Input +

All Input Conditions and Output Assertions are expressed in the same 6-element finite grammar

Output Assertions

i. This route installed in Adj-RIB-In shall be propagated as an outbound update to all eBGP peers.

ii. This route installed in Adj-RIB-In shall be propagated as an outbound update to all iBGP peers.

Output +

Logic is in an adjacent table

Items per page: 8 | 1–3 of 3 items

Scenarios

	A	B	C
✓	✓	X	X
X	X	✓	X
X	X	X	✓
!!	!!	!!	!!
!!	!!	!!	⊘

+

DWDS **RuleMaker**

Authoring app.

User sets the symbol icons for the logic of each scenario.

Controlled vocabulary and math can be expressed.

e.g. BGP-4 Dissemination Rule

I instructed https://venice.ai LLM to express the rule in “simple discrete declarative sentences using the finite grammar of the **Data With Direction Specification (DWDS)**”

Input Conditions

1. This route is received in Adj-RIB-In from an eBGP peer.
2. This route is received in Adj-RIB-In from a local service.
3. This route is received in Adj-RIB-In from an iBGP peer.

Input +

Output Assertions

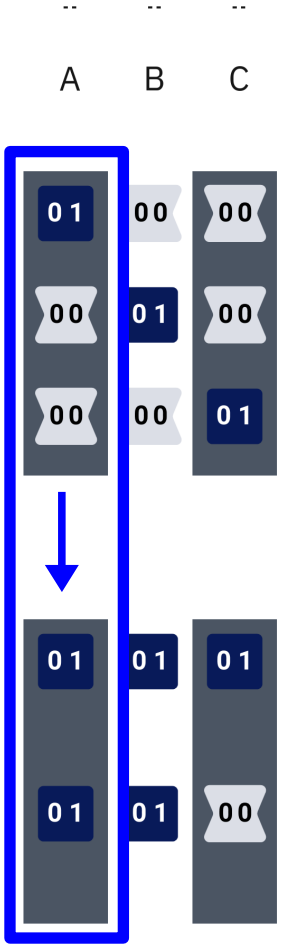
i. This route installed in Adj-RIB-In shall be propagated as an outbound update to all eBGP peers.

ii. This route installed in Adj-RIB-In shall be propagated as an outbound update to all iBGP peers.

Output +

Items per page: 8 | 1–3 of 3 items

Scenarios +



DWDS **RuleMaker**

Authoring App

User sets the {00,01,10,11} icons for the logic of each.

Export in DWD .txt (or .dwd) Rules-as-Data form, to IPFS node(s), local, or own database.

e.g. BGP-4 Dissemination Rule

I instructed https://venice.ai LLM to express the rule in “simple discrete declarative sentences using the finite grammar of the **Data With Direction Specification (DWDS)**”

Input Conditions

1. This route is received in Adj-RIB-In from an eBGP peer.
2. This route is received in Adj-RIB-In from a local service.
3. This route is received in Adj-RIB-In from an iBGP peer.

Input +

Output Assertions

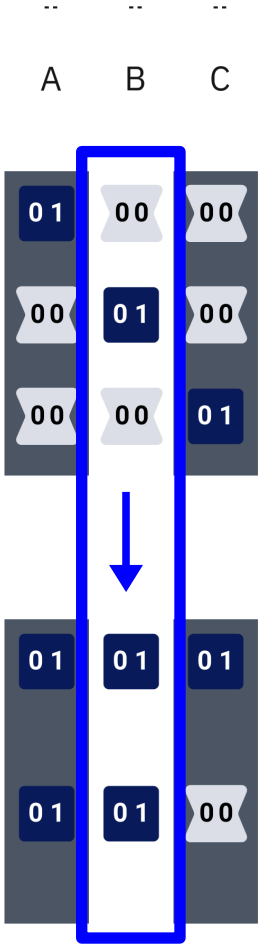
i. This route installed in Adj-RIB-In shall be propagated as an outbound update to all eBGP peers.

ii. This route installed in Adj-RIB-In shall be propagated as an outbound update to all iBGP peers.

Output +

Items per page: 8 | 1–3 of 3 items

Scenarios +



DWDS **RuleMaker**

Authoring App

User sets the {00,01,10,11} icons for the logic of each.

Export in DWD .txt (or .dwd) Rules-as-Data form, to IPFS node(s), local, or own database.

e.g. BGP-4 Dissemination Rule

I instructed https://venice.ai LLM to express the rule in “simple discrete declarative sentences using the finite grammar of the **Data With Direction Specification (DWDS)**”

Input Conditions

1. This route is received in Adj-RIB-In from an eBGP peer.
2. This route is received in Adj-RIB-In from a local service.
3. This route is received in Adj-RIB-In from an iBGP peer.

Input +

Output Assertions

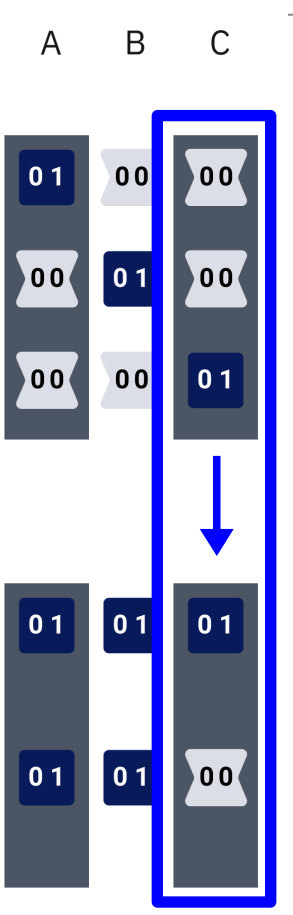
i. This route installed in Adj-RIB-In shall be propagated as an outbound update to all eBGP peers.

ii. This route installed in Adj-RIB-In shall be propagated as an outbound update to all iBGP peers.

Output +

Items per page: 8 | 1–3 of 3 items

Scenarios



DWDS **RuleMaker**

Authoring App

User sets the {00,01,10,11} icons for the logic of each.

Export in DWD .txt (or .dwd) Rules-as-Data form, to IPFS node(s), local, or own database.

Generic DWD Array

Transient ‘in memory’ representation.

Key-value sparse data structure.

A column-indexed table with:
- Key combinations at top, and
- Values in an identity array at bottom.

The structure of a three-dimensional "DWD KEY-VALUE Array" (DWD Array)

INDEX	1	2	3	4	5	6	7	8	9	10	11	12
KEY1	-	-	-	-	-	-	-	-	-	-	-	-
K1.1	a	1	0	0	1	0	0	1	0	0	1	0	0
K1.2	b	0	1	0	0	1	0	0	1	0	0	1	0
K1.3	c	0	0	1	0	0	1	0	0	1	0	0	1
KEY2	-	-	-	-	-	-	-	-	-	-	-	-	-
K2.1	i	1	1	1	1	1	1	0	0	0	0	0	0
K2.2	ii	0	0	0	0	0	0	1	1	1	1	1	1
KEY3	-	-	-	-	-	-	-	-	-	-	-	-	-
K3.1	ip	1	1	1	0	0	0	0	0	0	0	0	0
K3.2	iq	0	0	0	1	1	1	0	0	0	0	0	0
K3.3	iip	0	0	0	0	0	0	1	1	1	0	0	0
K3.4	iiq	0	0	0	0	0	0	0	0	0	1	1	1
VALUE	-	-	-	-	-	-	-	-	-	-	-	-	-
V_K1.1_K2.1_K3.1	aip	1	0	0	0	0	0	0	0	0	0	0	0
V_K1.2_K2.1_K3.1	bip	0	1	0	0	0	0	0	0	0	0	0	0
V_K1.3_K2.1_K3.1	cip	0	0	1	0	0	0	0	0	0	0	0	0
V_K1.1_K2.1_K3.2	aiq	0	0	0	1	0	0	0	0	0	0	0	0
V_K1.2_K2.1_K3.2	biq	0	0	0	0	1	0	0	0	0	0	0	0
V_K1.3_K2.1_K3.2	ciq	0	0	0	0	0	1	0	0	0	0	0	0
V_K1.1_K2.2_K3.3	aip	0	0	0	0	0	0	1	0	0	0	0	0
V_K1.2_K2.2_K3.3	bip	0	0	0	0	0	0	0	1	0	0	0	0
V_K1.3_K2.2_K3.3	cip	0	0	0	0	0	0	0	0	1	0	0	0
V_K1.1_K2.2_K3.4	aiq	0	0	0	0	0	0	0	0	0	1	0	0
V_K1.2_K2.2_K3.4	biq	0	0	0	0	0	0	0	0	0	0	1	0
V_K1.3_K2.2_K3.4	ciq	0	0	0	0	0	0	0	0	0	0	0	1

The structure of "DWD KEY-VALUE Coordinates List" (DWD Coordinates)

```
|ROW|COLUMN|
|KEY1|-|
|a|1,4,7,10|
|b|2,5,8,11|
|c|3,6,9,12|
|KEY2|-|
|i|1,2,3,4,5,6|
|ii|7,8,9,10,11,12|
|KEY3|-|
|ip|1,2,3|
|iq|4,5,6|
|iip|7,8,9|
|iiq|10,11,12|
|VALUE|-|
|aip|1|
|bip|2|
|cip|3|
|aiq|4|
|biq|5|
|ciq|6|
|aip|7|
|bip|8|
|cip|9|
|aiq|10|
|biq|11|
|ciq|12|
```

Generic DWD Coordinates

Expresses the same data as in the DWD Array:
- Redundant 0's removed, 1's replaced with column index numbers.
- Remains directly readable, supporting audit for data-integrity.

Persistent representation for storage and processing:
- Fast to process in **RuleTaker** (sift for "invoked" assertions.
- Minimal data to store in **RuleReserve** (one row per rule)
- Minimal data to transmit (Websocket or MCP)

A single consistent minimalist representation:
- LLMs can be trained to generate this directly from RFC paragraphs.
- Consistent API and tooling for rule use on any platform.
- Scales elegantly for complicated rules & large lookup tables.

BGP-4 Dissemination Rule DWD Coordinates

To express rule logic.

Produced in **RuleMaker.**

- LLMs aid in sentence drafting.
- humans refine & validate.
- DWD Coordinates autogenerated.

Stored in **RuleReserve.**

- with metadata.
- single line (one row per rule).
- decentralized and distributed.

```
INDEX|DATA|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY1|Scenarios|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY1.1|A|1|4|7|10|13|
|KEY1.2|B|2|5|8|11|14|
|KEY1.3|C|3|6|9|12|15|
|KEY2|Function|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY2.1|Input Condition|1|2|3|4|5|6|7|8|9|
|KEY2.2|Output Assertion|10|11|12|13|14|15|
|KEY3|Expression|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY3.1|{"determiner":"This","noun":"route","predicate_verb":"is",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from an eBGP peer."}|1|2|3|
|KEY3.2|{"determiner":"This","noun":"route","predicate_verb":"is",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from a local service."}|4|5|6|
|KEY3.3|{"determiner":"This","noun":"route","predicate_verb":"is ",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from an iBGP peer."}|7|8|9|
|KEY3.4|{"determiner":"  This ","noun":"route","past_participle_verb":"installed",
"attribute":"in Adj-RIB-In","predicate_verb":"shall be propagated as",
"description":"an outbound update to all eBGP peers."}|10|11|12|
|KEY3.5|{"determiner":"  This","noun":"route","past_participle_verb":"installed",
"attribute":"in Adj-RIB-In","predicate_verb":"shall be propagated as",
"description":"an outbound update to all iBGP peers. "}|13|14|15|
|V_K1.1_K2.1_K3.1|01|1|
|V_K1.2_K2.1_K3.1|00|2|
|V_K1.3_K2.1_K3.1|00|3|
|V_K1.1_K2.1_K3.2|00|4|
|V_K1.2_K2.1_K3.2|01|5|
|V_K1.3_K2.1_K3.2|00|6|
|V_K1.1_K2.1_K3.3|00|7|
|V_K1.2_K2.1_K3.3|00|8|
|V_K1.3_K2.1_K3.3|01|9|
|V_K1.1_K2.2_K3.4|01|10|
|V_K1.2_K2.2_K3.4|01|11|
|V_K1.3_K2.2_K3.4|01|12|
|V_K1.1_K2.2_K3.5|01|13|
|V_K1.2_K2.2_K3.5|01|14|
|V_K1.3_K2.2_K3.5|00|15|
```

BGP-4 Dissemination Rule Metadata

Produced in **RuleMaker**

LLMs used to help iteratively draft:
≤ 120-char rule title;
≤ 240-char summary; and
≤ 960-char explanation;
- humans refine & validate.

RuleTaker sends **is.dwd** message
- currently Websocket.
- soon to be MCP-enabled.

Rule discovered in **RuleReserve** using only the metadata:

- sifted for “in effect” (time; jurisd.)
- sifted for “applicable” (categories)
- sends **ought1.dwd** message:
  - one rule per row.
  - one lookup table per row.

```
|rule_id|8840e3db-57ec-4ccb-ac67-b51e6cc01790|
|rulereserve_nodes|*|
|version_standard_url|https://semver.org/|
|ruledata_version|0.0.0|
|properties.id|8840e3db-57ec-4ccb-ac67-b51e6cc01790|
|metadata.rule.120_title|Dissemination Rule of Border Gateway Protocol 4 (BGP-4)|
|metadata.rule.240_summary|BGP-4's Dissemination rule enables efficient routing by setting
how routes are shared. Routes learned from eBGP or locally are sent to all peers.
iBGP-learned routes are sent only to eBGP peers. This enures full a mesh and prevents loops.|
|metadata.rule.960_explanation|BGP-4's Dissemination rules govern how routing information is
shared between autonomous systems (AS) for efficient data transmission. Routes learned from
external BGP (eBGP) peers or originated locally are propagated to all peers, ensuring
widespread awareness of new paths. In contrast, routes learned from internal BGP (iBGP) peers
are only sent to eBGP peers, maintaining a full mesh topology within the AS to prevent routing
loops. This structured approach optimizes inter-AS communication and avoids inefficiencies,
aking BGP-4 essential for the internet's functioning.|
|metadata.rule.rule_group||
|metadata.rule.rule_version||
|metadata.rule.rule_criticality|experimental|
|metadata.rule.url|https://www.rfc-editor.org/rfc/rfc4271|
|metadata.rule.pattern||
|metadata.rule.pattern_version||
|metadata.rule.rulemaker_entity.1.name|Internet Engineering Task Force|
|metadata.rule.rulemaker_entity.1.url|https://www.ietf.org/|
|metadata.rule.rulemaker_entity.1.uuid|c66c4845-6592-461f-bb09-2ba19da12cc4|
|metadata.rule.rulemaker_manager.1.name|Y. Rekhter|
|metadata.rule.rulemaker_manager.1.email|yakov@watson.ibm.com|
|metadata.rule.rulemaker_manager.1.contact|Editor, Network Working Group|
|metadata.rule.rulemaker_manager.1.uuid|4a13b6bd-645b-46a7-9528-f8246aaa68e8|
|metadata.rule.version||
|linked_rules_or_lookups|[]|
|in_effect.1.country|ALL|
|in_effect.1.subcountry||
|in_effect.1.timezone|2025-07-07T04:49:51-12:00|
|in_effect.1.afterDate|2004-12-31T05:00:01.000Z|
|in_effect.1.untilDate|2023-12-30T05:00:01.000Z|
|category_applicable.industry.1.isic_code|6190|
|category_applicable.industry.1.isic_name|Other telecommunications activities|
|category_applicable.good_service_asset.1.unspsc_code| 43222641|
|category_applicable.good_service_asset.1.unspsc_name|Internet protocol sharing device|
|qualitative_weight.character|88|
|qualitative_weight.enforcement|9|
|qualitative_weight.consequences|89|
|normative_orientation.rulemaker_ruletaker_thirdparty|rule-taker|
|normative_orientation.affirmative_negative_interrogative|have|
|normative_orientation.be_do_have|do|
|normative_orientation.declarative_imperative_empirical|declarative|
|normative_orientation.logical_practical_ethical|logical|
|normative_orientation.must_may_should|must|
```

BGP-4 Dissemination Rule DWD Coordinates

Received by **RuleTaker**

- as an **ought1.dwd** message.

- uses local circumstantial data to sift DWD Coordinates for all the “invoked” Output Assertions.

- structures an **ought2.dwd** record in memory, containing only the “invoked” Output Assertions.

- Ready for any substrate application:
  - display to a human.
  - OR run ‘as is’ with a parser.
  - OR transcribe to any code by a human or an LLM, to run in any app or rules engine.

```
INDEX|DATA|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY1|Scenarios|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY1.1|A|1|4|7|10|13|
|KEY1.2|B|2|5|8|11|14|
|KEY1.3|C|3|6|9|12|15|
|KEY2|Function|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY2.1|Input Condition|1|2|3|4|5|6|7|8|9|
|KEY2.2|Output Assertion|10|11|12|13|14|15|
|KEY3|Expression|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
|KEY3.1|{"determiner":"This","noun":"route","predicate_verb":"is",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from an eBGP peer."}|1|2|3|
|KEY3.2|{"determiner":"This","noun":"route","predicate_verb":"is",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from a local service."}|4|5|6|
|KEY3.3|{"determiner":"This","noun":"route","predicate_verb":"is ",
"past_participle_verb":"received","description":"in Adj-RIB-In",
"attribute":"from an iBGP peer."}|7|8|9|
|KEY3.4|{"determiner":"  This ","noun":"route","past_participle_verb":"installed",
"attribute":"in Adj-RIB-In","predicate_verb":"shall be propagated as",
"description":"an outbound update to all eBGP peers."}|10|11|12|
|KEY3.5|{"determiner":"  This","noun":"route","past_participle_verb":"installed",
"attribute":"in Adj-RIB-In","predicate_verb":"shall be propagated as",
"description":"an outbound update to all iBGP peers. "}|13|14|15|
|V_K1.1_K2.1_K3.1|01|1|
|V_K1.2_K2.1_K3.1|00|2|
|V_K1.3_K2.1_K3.1|00|3|
|V_K1.1_K2.1_K3.2|00|4|
|V_K1.2_K2.1_K3.2|01|5|
|V_K1.3_K2.1_K3.2|00|6|
|V_K1.1_K2.1_K3.3|00|7|
|V_K1.2_K2.1_K3.3|00|8|
|V_K1.3_K2.1_K3.3|01|9|
|V_K1.1_K2.2_K3.4|01|10|
|V_K1.2_K2.2_K3.4|01|11|
|V_K1.3_K2.2_K3.4|01|12|
|V_K1.1_K2.2_K3.5|01|13|
|V_K1.2_K2.2_K3.5|01|14|
|V_K1.3_K2.2_K3.5|00|15|
```

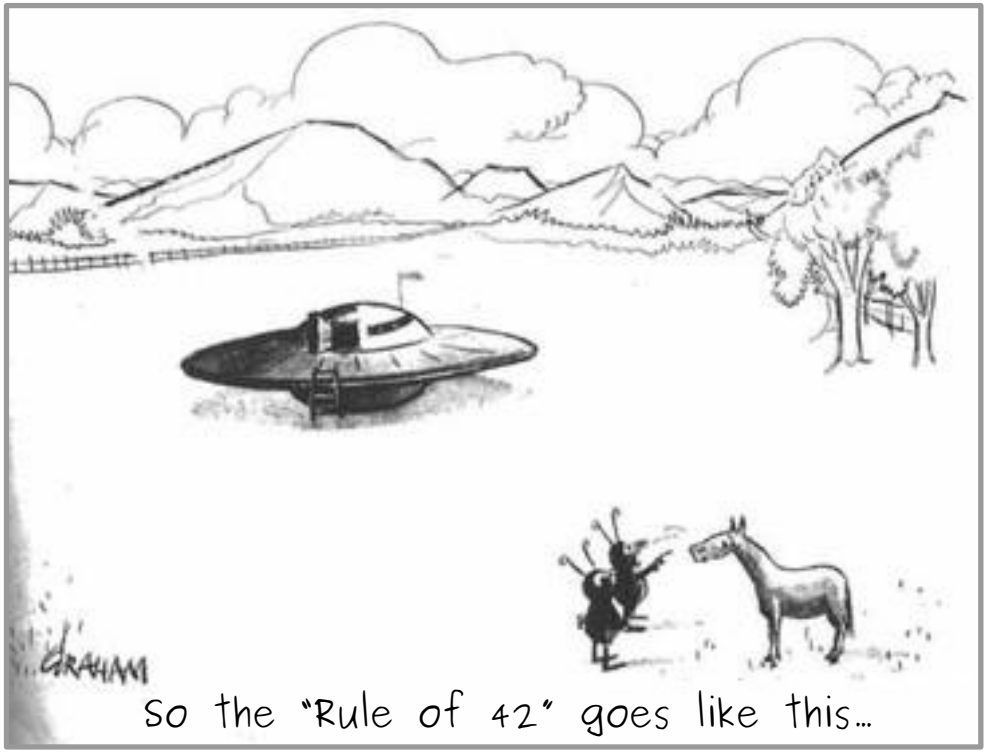
... the elements of an Internet of Rules?

https://xalgorithms.org

https://gitlab.com/xalgorithms-alliance
joseph_potvin on GitLab

https://discord.gg/zvDjzHGE
josephpotvin on Discord

jpotvin@xalgorithms.org
(819) 593-5983 on SMS, Signal, Telegram
@xalgorithms on X



Adapted from the original 1953 cartoon by Alex Graham in *The New Yorker* magazine.

Joseph Potvin
2025-11-05

IRTF

Xalgorithms Foundation

THE OPMAN COMPANY

Université du Québec en Outaouais